

1      **CLAIMS**

2      What is claimed, is:

3      (1) A compiler apparatus for collecting frequencies with which  
4      each process is executed in a program to be optimized and  
5      optimizing said program based on the collected frequencies,  
6      said apparatus comprising:

7      a loop process detection portion for detecting a repeatedly  
8      executed loop process of said program;

9      a loop process frequency collection portion for collecting loop  
10     process frequencies with which said loop process is executed in  
11     said program;

12     an in-loop process frequency collection portion for collecting  
13     in-loop process frequencies with which, as against the number  
14     of times of execution of said loop process, each of a plurality  
15     of in-loop processes included in said loop process is executed;

16     an in-loop execution information generating portion for, based  
17     on said loop process frequencies and said in-loop process  
18     frequencies, generating in-loop execution information  
19     indicating the frequencies with which each of said plurality of  
20     in-loop processes is executed in the case where said program is  
21     executed; and

22     an optimization portion for optimizing said program based on  
23     said in-loop execution information generated by said in-loop  
24     execution information generating portion.

25     (2) The compiler apparatus according to Claim 1, wherein said  
26     in-loop process frequency collection portion collects said

1 in-loop process frequencies in the case where said loop process  
2 frequencies are higher than a predetermined frequency.

3 (3) The compiler apparatus according to Claim 1, wherein said  
4 in-loop execution information generating portion generates said  
5 in-loop execution information by multiplying said loop process  
6 frequencies by said in-loop process frequencies.

7 (4) The compiler apparatus according to Claim 1, wherein:

8 said loop process is an outer loop process including an inner  
9 loop process which is a further inside loop process;

10 said loop process detection portion further detects said inner  
11 loop process;

12 said loop process frequency collection portion collects the  
13 loop process frequencies with which said inner loop process is  
14 executed in said program based on said in-loop execution  
15 information;

16 said in-loop process frequency collection portion further  
17 collects the in-loop process frequencies of said inner loop  
18 process; and

19 said in-loop execution information generating portion generates  
20 the in-loop execution information on said inner loop process by  
21 multiplying the in-loop process frequencies in said inner loop  
22 process by said loop process frequencies of said inner loop  
23 process.

24 (5) The compiler apparatus according to Claim 1, wherein:

25 said loop process frequency collection portion stops a counter

1 for determining the number of times of execution of said loop  
2 process when said program is executed a predetermined number of  
3 times so as to collect the number of times determined by the  
4 counter as said loop process frequencies; and

5 said in-loop process frequency collection portion stops the  
6 counter for determining the number of times of execution of  
7 each of said plurality of in-loop processes when a total of  
8 determined values of said plurality of in-loop processes  
9 becomes the predetermined number of times.

10 (6) The compiler apparatus according to Claim 1, further  
11 comprising:

12 a control flow graph generating portion for generating a  
13 control flow graph in which each of a plurality of instruction  
14 sequences in said program is generated as a node and an  
15 execution order of said plurality of instruction sequences is  
16 generated as a directed edge of said nodes;

17 a structure graph generating portion for, in said control flow  
18 graph, generating an outline structure graph in which a single  
19 loop node for showing said loop process in its entirety is  
20 generated instead of a collection of the nodes forming said  
21 loop process and an in-loop structure graph which is the  
22 control flow graph of the collection of the nodes forming said  
23 loop process; and

24 a counter insertion portion for, in each of said outline  
25 structure graph and said in-loop structure graph, inserting a  
26 counter into said program in order to count the number of times  
27 of execution of each execution path in the structure graphs,  
28 and wherein:

1       said loop process frequency collection portion generates as  
2       said loop process frequencies the numbers of times of execution  
3       of said loop node as against the numbers of times of execution  
4       of said program; and

5       said in-loop process frequency collection portion collects as  
6       said in-loop process frequencies the number of times of  
7       execution of each execution path in said in-loop structure  
8       graph as against the numbers of times of execution of said loop  
9       process.

10      (7) The compiler apparatus according to Claim 6, wherein:

11     in the case where said program is executed a predetermined  
12     number of times, said loop process frequency collection portion  
13     collects as the loop process frequencies the determined values  
14     of the counter inserted for counting the number of times of  
15     execution of the execution paths including said loop node; and

16     in the case where a total of the determined values of said  
17     plurality of in-loop processes becomes a predetermined number  
18     of times, said in-loop process frequency collection portion  
19     collects the in-loop process frequencies based on the  
20     determined values of the counter inserted for counting the  
21     number of times of execution of each execution path in said  
22     in-loop structure graph.

23      (8) The compiler apparatus according to Claim 6, wherein in the  
24     case where an insertion position in said program for inserting  
25     the counter for determining the number of times of execution of  
26     each execution path in said outline structure graph is the same  
27     as the position in said program for inserting the counter for  
28     determining the number of times of execution of each execution  
29     path in said in-loop structure graph and then the counter of

1 one, at the most, of said outline structure graph and said  
2 in-loop structure graph is started, said counter insertion  
3 portion inserts into the insertion position the counter for  
4 determining the numbers of times of execution of the execution  
5 paths in both said outline structure graph and said in-loop  
6 structure graph.

7 (9) The compiler apparatus according to Claim 6, wherein:

8 in the case where an insertion position in said program for  
9 inserting the counter for determining the number of times of  
10 execution of each execution path in said outline structure  
11 graph is the same as the position in said program for inserting  
12 the counter for determining the number of times of execution of  
13 each execution path in said in-loop structure graph and then  
14 the counter of one, at the most, of said outline structure  
15 graph and said in-loop structure graph is started, said counter  
16 insertion portion generates a plurality of determination  
17 processes for determining the number of times of execution of  
18 each execution path in each of said outline structure graph and  
19 said in-loop structure graph; and

20 said in-loop process frequency collection portion inserts a  
21 jump instruction for moving the process to another portion into  
22 said insertion position and sets a jump destination of the jump  
23 instruction at one of said plurality of determination processes  
24 so as to determine the numbers of times of execution of the  
25 execution paths in both said outline structure graph and said  
26 in-loop structure graph.

27 (10) The compiler apparatus according to Claim 6, wherein:

28 said loop process is an outer loop process including an inner  
29 loop process which is a further inside loop process;

1       said loop process detection portion further detects said inner  
2       loop process;

3       in the control flow graph of said outer loop process, said  
4       structure graph generating portion generates as an in-outer  
5       loop structure graph a graph in which the single inner loop  
6       node is generated instead of a collection of the nodes forming  
7       said inner loop process and generates an in-inner loop  
8       structure graph which is the control flow graph of the  
9       collection of the nodes forming said inner loop process; and  
10      said counter insertion portion further inserts the counter for  
11      determining the number of times of execution of each execution  
12      path in the in-inner loop structure graph;

13      said loop process frequency collection portion further collects  
14      the loop process frequencies with which said inner loop process  
15      is executed in said program based on said in-loop execution  
16      information;

17      said in-loop process frequency collection portion collects the  
18      frequencies of execution of each execution path in said  
19      in-inner loop structure graph as the in-loop process  
20      frequencies of said inner loop process as against the number of  
21      times of execution of said inner loop process; and

22      said in-loop execution information generating portion further  
23      generates the in-loop execution information on said inner loop  
24      process by multiplying the in-loop process frequencies in said  
25      inner loop process by the loop process frequencies of said  
26      inner loop process.

27      (11) The compiler apparatus according to Claim 10, wherein, in  
28      the case where an insertion position in said program for

1 inserting the counter for determining the number of times of  
2 execution of each execution path in said in-outer loop  
3 structure graph is the same as the position in said program for  
4 inserting the counter for determining the number of times of  
5 execution of each execution path in said in-inner loop  
6 structure graph and then the counter of one, at the most, of  
7 said in-outer loop structure graph and said in-inner loop  
8 structure graph is started, said counter insertion portion  
9 inserts into the insertion position the counter for determining  
10 the numbers of times of execution of the execution paths in  
11 both said in-outer loop structure graph and said in-inner loop  
12 structure graph.

13 (12) The compiler apparatus according to Claim 10, wherein:  
14 in the case where an insertion position in said program for  
15 inserting the counter for determining the number of times of  
16 execution of each execution path in said in-outer loop  
17 structure graph is the same as the position in said program for  
18 inserting the counter for determining the number of times of  
19 execution of each execution path in said in-inner loop  
20 structure graph and then the counter of one, at the most, of  
21 said in-outer loop structure graph and said in-inner loop  
22 structure graph is started, said counter insertion portion  
23 generates a plurality of determination processes for  
24 determining the number of times of execution of each execution  
25 path in each of said in-outer loop structure graph and said  
26 in-inner loop structure graph; and  
27 said in-loop process frequency collection portion inserts a  
28 jump instruction for moving the process to another portion into  
29 said insertion position and sets a jump destination of the jump  
30 instruction at one of said plurality of determination processes  
31 so as to determine the number of times of execution of the

1 execution paths in both said in-outer loop structure graph and  
2 said in-inner loop structure graph.

3 (13) A compiler program for causing a computer to function as  
4 a compiler apparatus for collecting frequencies with which each  
5 process is executed in a program to be optimized and optimizing  
6 said program based on the collected frequencies, said program  
7 causing said computer to function as:

8 a loop process detection portion for detecting a repeatedly  
9 executed loop process of said program;

10 a loop process frequency collection portion for collecting loop  
11 process frequencies with which said loop process is executed in  
12 said program;

13 an in-loop process frequency collection portion for collecting  
14 in-loop process frequencies with which, as against the number  
15 of times of execution of said loop process, each of a plurality  
16 of in-loop processes included in said loop process is executed;

17 an in-loop execution information generating portion for, based  
18 on said loop process frequencies and said in-loop process  
19 frequencies, generating in-loop execution information  
20 indicating the frequencies with which each of said plurality of  
21 in-loop processes is executed in the case where said program is  
22 executed; and

23 an optimization portion for optimizing said program based on  
24 said in-loop execution information generated by said in-loop  
25 execution information generating portion.

26 (14) The record medium having the compiler program according  
27 to Claim 13 recorded thereon.

1       (15) A compilation method for collecting frequencies with  
2       which each process is executed in a program to be optimized and  
3       optimizing said program based on the collected frequencies,  
4       said method having:  
5       a loop process detection step of detecting a repeatedly  
6       executed loop process of said program;  
7       a loop process frequency collection step of collecting loop  
8       process frequencies with which said loop process is executed in  
9       said program;  
10      an in-loop process frequency collection step of collecting  
11      in-loop process frequencies with which, as against the number  
12      of times of execution of said loop process, each of a plurality  
13      of in-loop processes included in said loop process is executed;  
14      an in-loop execution information generating step of, based on  
15      said loop process frequencies and said in-loop process  
16      frequencies, generating in-loop execution information  
17      indicating the frequencies with which each of said plurality of  
18      in-loop processes is executed in the case where said program is  
19      executed; and  
20      an optimization step of optimizing said program based on said  
21      in-loop execution information generated by said in-loop  
22      execution information generating portion.  
23      (16) A runtime information generating apparatus for collecting  
24      frequencies with which each process is executed in a program to  
25      be optimized, said apparatus having:  
26      a loop process detection portion for detecting a repeatedly

1       executed loop process of said program;

2       a loop process frequency collection portion for collecting loop  
3       process frequencies with which said loop process is executed in  
4       said program;

5       an in-loop process frequency collection portion for collecting  
6       in-loop process frequencies with which, as against the number  
7       of times of execution of said loop process, each of a plurality  
8       of in-loop processes included in said loop process is executed;

9       an in-loop execution information generating portion for, based  
10      on said loop process frequencies and said in-loop process  
11      frequencies, generating in-loop execution information  
12      indicating the frequencies with which each of said plurality of  
13      in-loop processes is executed in the case where said program is  
14      executed, and

15      optimizing said program based on said in-loop execution  
16      information generated by said in-loop execution information  
17      generating portion.

18      (17) A runtime information generating program for causing a  
19      computer to function as a runtime information generating  
20      apparatus for collecting frequencies with which each process is  
21      executed in a program to be optimized, said program causing  
22      said computer to function as:

23      a loop process detection portion for detecting a repeatedly  
24      executed loop process of said program;

25      a loop process frequency collection portion for collecting loop  
26      process frequencies with which said loop process is executed in  
27      said program;

1 an in-loop process frequency collection portion for collecting  
2 in-loop process frequencies with which, as against the number  
3 of times of execution of said loop process, each of a plurality  
4 of in-loop processes included in said loop process is executed;  
5 and

6 an in-loop execution information generating portion for, based  
7 on said loop process frequencies and said in-loop process  
8 frequencies, generating in-loop execution information  
9 indicating the frequencies with which each of said plurality of  
10 in-loop processes is executed in the case where said program is  
11 executed, and

12 causing said program to be optimized based on said in-loop  
13 execution information generated by said in-loop execution  
14 information generating portion.

15 (18) A record medium having a runtime information generating  
16 program according to Claim 17 recorded thereon.

17 (19) A computer program product comprising a computer usable  
18 medium having computer readable program code means embodied  
19 therein for causing collection of frequencies with which each  
20 process is executed in a program to be optimized, the computer  
21 readable program code means in said computer program product  
22 comprising computer readable program code means for causing a  
23 computer to effect the functions of claim 1.

24 (20) An article of manufacture comprising a computer usable  
25 medium having computer readable program code means embodied  
26 therein for causing collection of frequencies with which each  
27 process is executed in a program to be optimized, the computer  
28 readable program code means in said article of manufacture

1 comprising computer readable program code means for causing a  
2 computer to effect the steps of claim 15.

3 (21) A program storage device readable by machine, tangibly  
4 embodying a program of instructions executable by the machine  
5 to perform method steps for collecting frequencies with which  
6 each process is executed in a program to be optimized, said  
7 method steps comprising the steps of claim 15.

8 (22) A computer program product comprising a computer usable  
9 medium having computer readable program code means embodied  
10 therein for causing collection of frequencies with which each  
11 process is executed in a program to be optimized, the computer  
12 readable program code means in said computer program product  
13 comprising computer readable program code means for causing a  
14 computer to effect the functions of claim 16.

15 (23) A runtime information generating method comprising:

16 collecting frequencies with which each process is executed in a  
17 program to be optimized, said step of collecting frequencies  
18 comprising:

19 detecting a repeatedly executed loop process of said program;

20 collecting loop process frequencies with which said loop  
21 process is executed in said program;

22 collecting in-loop process frequencies with which, as against  
23 the number of times of execution of said loop process, each of  
24 a plurality of in-loop processes included in said loop process  
25 is executed;

26 based on said loop process frequencies and said in-loop process

1 frequencies, generating in-loop execution information  
2 indicating the frequencies with which each of said plurality of  
3 in-loop processes is executed in the case where said program is  
4 executed, and

5 optimizing said program based on said in-loop execution  
6 information generated by said in-loop execution information  
7 generating portion.

8 (24) An article of manufacture comprising a computer usable  
9 medium having computer readable program code means embodied  
10 therein for causing runtime information generation, the  
11 computer readable program code means in said article of  
12 manufacture comprising computer readable program code means for  
13 causing a computer to effect the steps of claim 23.

14 (25) A program storage device readable by machine, tangibly  
15 embodying a program of instructions executable by the machine  
16 to perform method steps for runtime information generation,  
17 said method steps comprising the steps of claim 15.